

Spajanje tabela

Dekartov proizvod

CROSS JOIN

U rezultujućoj tabeli, svaki red iz prve tabele je spojen sa svakim redom iz druge tabele.

```
SELECT *  
FROM games CROSS JOIN devs
```

Ovakvo spajanje tabela je podrazumevano u SQL-u, zato možemo izostaviti **CROSS JOIN**, radiće identično.

```
SELECT *  
FROM games, devs
```

Filtriranje

WHERE

Kada dobijemo dekartov proizvod, dodatno možemo filtrirati redove na standardan način, pomoću **WHERE**

```
SELECT *  
FROM games, devs  
WHERE games.dev_id = devs.dev_id
```

Ovakav uslov daje utisak da smo spajanje ograničili na redove koji imaju istu vrednost za date kolone, ali smo prethodno zapravo dobili celokupan dekartov proizvod i onda naknadno isfiltrirali redove po datom uslovu.

Uslovi u **WHERE** naravno mogu biti kompleksniji

```
SELECT *  
FROM games, devs  
WHERE games.dev_id = devs.dev_id  
AND games.reviews > 99  
AND games.year < 2019
```

Alias

Nazive rezultujućih kolona možemo promeniti

```
SELECT games.title AS "Igrica", devs.name AS "Studio"  
FROM games, devs  
WHERE games.dev_id = devs.dev_id
```

Takođe, za potrebe upita, možemo preimenovati tabele koje učestvuju u upitu

```
SELECT g.title, d.name  
FROM games g, devs d  
WHERE g.dev_id = d.dev_id
```

Pojmovi:

- *Equijoin* - Spajanje pomoću filtera koji poredi jednakost kolona (=)
- *Non-Equijoin* - Spajanje pomoću bilo kog drugog filtera

Inner Join

JOIN / INNER JOIN

On

Umesto da spajamo sve redove pomoću dekartovog proizvoda pa da naknadno radimo filtriranje, možemo odmah navesti uslov po kome će se redovi spajati.

```
SELECT *  
FROM games JOIN devs  
ON games.dev_id = devs.dev_id
```

Nakon spajanja pomoću uslova u **ON** delu, možemo odraditi klasično filtriranje pomoću **WHERE**:

```
SELECT *  
FROM games JOIN devs  
ON games.dev_id = devs.dev_id  
WHERE games.reviews > 99  
AND games.year < 2019
```

Using

U slučaju da se kolone kojima upoređujemo jednakost (*equijoin*) prilikom spajanja zovu isto, možemo koristiti **USING** ključnu reč:

```
SELECT *  
FROM games JOIN devs  
USING dev_id
```

Natural Join

NATURAL JOIN

Slično naredbi **USING**, **NATURAL JOIN** automatski spaja tabele po jednakosti kolona, i to automatski za sve kolone koje se u datim tabelama zovu isto.

```
SELECT *  
FROM games NATURAL JOIN devs
```

Spajanje više od dve tabele

Spajanje se uvek izvršava za neke **dve** tabele. Kada se dve tabele spoje dobijemo jednu rezultujuću tabelu koju onda možemo spojiti sa **trećom**, što će opet dati novu rezultujuću tabelu koju onda možemo spojiti sa **četvrtom**, i tako dalje.

```
SELECT *  
FROM games  
JOIN devs USING dev_id  
JOIN cities ON devs.city_id = cities.id
```

Self Join

Spajanje redova iz tabele sa redovima iz te iste tabele.

Inner join sa uslovom za spajanje:

```
SELECT emp.last_name AS "Radnik",  
       mng.last_name AS "Rukovodilac"  
FROM employees emp JOIN employees mng  
ON emp.manager_id = mng.employee_id
```

Outer Join

Ukoliko se na osnovu uslova spajanja za neki red ne pronađe odgovarajući par u drugoj tabeli on može biti uključen u finalnu tabelu. U tom slučaju umesto vrednosti reda iz druge tabele (pošto nije pronađen) će se naći **NULL**.

Left Outer Join

LEFT JOIN / LEFT OUTER JOIN

Svi redovi **leve** tabele će biti uključeni u rezultat. Oni koji nemaju odgovarajući red iz **desne** tabele će za vrednosti kolona iz te tabele imati **NULL**.

```
SELECT w.name, m.name
FROM workers w LEFT JOIN managers m
ON w.manager_id = m.id
```

worker	manager
Pera	Žika
Mika	Žika
Laza	-

Right Outer Join

RIGHT JOIN / RIGHT OUTER JOIN

Svi redovi **desne** tabele će biti uključeni u rezultat. Oni koji nemaju odgovarajući red iz **leve** tabele će za vrednosti kolona iz te tabele imati **NULL**.

```
SELECT w.name, m.name
FROM workers w RIGHT JOIN managers m
ON w.manager_id = m.id
```

worker	manager
Pera	Žika
Mika	Žika
-	Raja

Full Outer Join

FULL JOIN / FULL OUTER JOIN

Svi redovi **obe** tabele će biti uključeni u rezultat. Oni koji nemaju odgovarajući red iz **druge** tabele će za vrednosti kolona iz te tabele imati **NULL**.

```
SELECT w.name, m.name
FROM workers w FULL JOIN managers m
ON w.manager_id = m.id
```

worker	manager
Pera	Žika
Mika	Žika
Laza	-
-	Raja

Oracle Syntax (+)

Starija Oracle-ova sintaksa za outer join nam omogućava da preko filtriranja dekartovog proizvoda efektivno odradimo **left/right outer join**.

Left Outer Join:

```
SELECT w.name, m.name
FROM workers w, managers m
WHERE w.manager_id(+) = m.id
```

Right Outer Join:

```
SELECT w.name, m.name
FROM workers w, managers m
WHERE w.manager_id = m.id(+)
```

Umesto da filtriranjem samo izbaci redove koji ne odgovaraju uslovu, određene redove će zadržati (u zavisnosti da li je **left** ili **right** join) ali će ih modifikovati tako da u odgovarajućim kolonama imaju **NULL**.

Hijerarhijski upiti ???
